1

AFIT/EN-TR-89-2

Air Force Institute of Technology

FAST STICK: A COMPUTER ASSISTED TACTICAL AIR EMPLOYMENT EXERCISE

Swen A. Walker   Mark A. Roth
Capt, USAF   Maj, USAF

June 5, 1989

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY

AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

1

UNCLASSIFIED

AFIT/EN-TR-89-2

Air Force Institute of Technology

FAST STICK: A COMPUTER ASSISTED TACTICAL AIR EMPLOYMENT EXERCISE

Swen A. Walker   Mark A. Roth
Capt, USAF   Maj, USAF

June 5, 1989

Approved for public release; distribution unlimited

FAST STICK: A Computer-Assisted Tactical Air Employment Exercise

Swen A. Walker, Capt, USAF*
Mark A. Roth, Maj, USAF

Air Force Institute of Technology
Department of Electrical and Computer Engineering (AFIT/ENG)
Wright-Patterson AFB, OH 45433-6583

Abstract

FAST STICK is a computer simulated tactical air employment exercise which serves as the capstone for the theater warfare phase of the Air Force's Air Command and Staff College curriculum. Its main objective is to provide intermediate level Air Force staff officers the opportunity to apply the basic tactical employment concepts of reconnaissance, counter air, interdiction, and close air support. The Air Force Institute of Technology identified several shortcomings of the exercise and proceeded to correct them. This paper describes the exercise, the addition of a scripted scenario generator for land play, and the replacement of the application dependent file structures with a database management system.

1 Introduction

FAST STICK is a computer simulated tactical air employment exercise which serves as the capstone for the theater warfare phase of the Air Force's Air Command and Staff College curriculum [1]. Its main objective is to provide intermediate level Air Force staff officers the opportunity to apply the basic tactical employment concepts of reconnaissance, counter air, interdiction, and close air support. The game is the final step in a sequence of exercises to deploy and employ air forces against the forces and targets of an imaginary enemy.

FAST STICK was originally written in Fortran and ran on a Honeywell H6000 mainframe computer. Input and output to the program was accomplished over a 300 baud hard copy terminal device. The interface was very user unfriendly. Input into the program had to follow a rigid format. Players would spend more time learning the computer syntax required to input data than learning to play the game. The program was also very inflexible. Changes could not be easily made to the game's scenario or parameters without a major rewrite of the code each time.

In August of 1987, the staff of Air Force Wargaming Center began a rapid prototyping effort to improve FAST STICK by rehosting it on a Zenith 158 microcomputer and modifying the user interface. The new simulation program was written in Pascal, and the user interface was replaced by a screen oriented menu driven system. Although major improvements were made to the program, the game had shortcomings as a joint exercise.

*Capt Walker is currently with the 1912 Computer Systems Group, Langley AFB, VA 23665-6349.

In this paper, we will present background on the FAST STICK exercise, an overview of the simulation model used, and two of the shortcomings of the game. This paper also contains our approach for redesigning the game to include land play in the simulation model, and our approach for modifying the application dependent file structure with a database management system.

2 Game Play

The FAST STICK game attempts to simulate the environment that an individual would be exposed to in the plans and operations branches of a Tactical Air Control Center (TACC) during a tactical war. In the exercise, individuals are members of a team that make up the TACC branches. Each team member performs a staff function of one of these branches. During the exercise, team members determine the priority of targets to be destroyed, assign a desired damage expectancy for each target, plan reconnaissance missions to obtain more information, and then decide on which targets to attack.

Before the game is started, a team is briefed on the general scenario of the game. The scenario basically depicts two fictitious countries with equal offensive and defense capabilities with open hostilities. The computer game is programmed with these capabilities and represents one of these countries, while a team represents the other. A team starts with a limited number of aircraft resources and is expected to meet stated objectives from higher command directives.

The FAST STICK simulation is played over four calendar days, while the game itself simulates only three days of actual events. On the first calendar day, a team will plan and conduct reconnaissance missions to obtain more information on enemy targets. On the second calendar day, a team reviews the reconnaissance data from game day one to plan attack and reconnaissance missions for game days two and three. Game days two and three are played on the third and fourth calendar days. On these days, a team plans and conducts attack and reconnaissance missions. At the end of the fourth day, the FAST STICK program compiles a score for the team based upon the number of targets destroyed, and the remaining aircraft resources.

All planning and flying is based on two cycles: morning and afternoon. A typical game day computer sequence of events would occur as follows [1, page 3-3]:

- Morning Cycle
  1. Logon to the game.
  2. Reserve air defense aircraft.
  3. Reserve ground spare aircraft.
  4. Reserve close air support aircraft.
  5. Input flight plans with takeoff times from 0001 to 1159.
  6. Engage simulation.
  7. Receive results from flights that recover prior to 1200.
- Noon – Take 30 to 40 minutes to assess the morning results and make any changes or additions to afternoon flight plans.
- Afternoon Cycle

1. Logon to the game.
2. Reserve air defense aircraft.
3. Reserve ground spare aircraft.
4. Reserve close air support aircraft.
5. Input flight plans with takeoff times from 1201 to 2359.
6. Engage simulation.
7. Receive results from flights that recover prior to 2400.

- At the end of the day, the program provides a summary of the day's activities.

3 The FAST STICK Computer Simulation Model

The FAST STICK simulation game is event driven. An example of an event would be an aircraft taking off from an airbase. For each type of mission found in the simulation program, a predetermined sequence of events exists. Each of these events has either a fixed or variable probability of success. In the model, the probability of success of an event is compared against a random number generated between zero and one by a Monte Carlo random number generation process. If the number generated is less than or equal to the probability of the event, the event is a success, and the program moves to the next event in the mission sequence. In order for a mission to be successful each event in the mission sequence must be successfully completed.

There are basically two types of missions found in the model, reconnaissance and attack. However, deviations from planned events within these two types of missions are possible. The difference between the two lies primarily in the number of events and the variable probabilities of success. Attack missions for the most part have more events in their mission sequence with a considerable number of these events having variable probabilities of success. This is a result of the fact that attack missions may use additional support forces.

An example reconnaissance mission sequence is shown in Figure 1. In the diagram, the first event that occurs on a reconnaissance mission is takeoff. The .95 in the box represents the probability of a successful takeoff. Should the takeoff be unsuccessful the aircraft is not damaged but will be unavailable for eight hours due to aircraft maintenance. The next event that may occur in the mission is air refueling. This event is dependent on whether a team has scheduled refueling in their flight plan. If air refueling was scheduled then the probability of successfully refueling is .98. If air refueling was missed then the aircraft attempts to return to base, otherwise the aircraft proceeds on with its mission. The next event to occur in the mission sequence is the aircraft flying over the target taking photographs. The probability of the reconnaissance flight surviving while over the target is .98. Should the aircraft be hit by enemy defenses while over the target, it will attempt to return to base and land. The probability of a successful landing in such a case is .60. If the aircraft lands, it will undergo maintenance. In the maintenance event, the aircraft will have a .10 chance of being in maintenance for 8 hours, a .40 chance of being in maintenance for 24 hours, and a .50 chance of being permanently damaged. Should the aircraft fail to land it is considered destroyed.

If the aircraft was not hit while photographing the target, then the next event determines whether useful intelligence information was photographed. This event has a variable probability based on the weather conditions over the target. As the aircraft begins its return trek, it may be scheduled for air refueling. This is dependent on a team's flight plan. The air refueling event

has the same probability of success and follows the same sequence as mentioned above. However, should the aircraft miss refueling with a tanker, the aircraft may not have enough fuel to return to base. If the aircraft cannot reach it's home base or a friendly base with it's remaining fuel, it crashes. Following refueling, the activity of returning to base and successfully landing is the next event. If this event is unsuccessful the aircraft is considered destroyed. The final event in the reconnaissance event sequence is a maintenance check to determine whether or not the aircraft will remain in commission and be available for additional missions. The probability of success is .80. Should the aircraft fail the maintenance check, it has a .40 chance of being in maintenance for 8 hours or a .60 chance of being in maintenance for 24 hours.

4 Shortcomings of the Game

The primary problem with the new simulation program is that there is no land or battle simulation play in the game. One of the objectives of this game is to teach Air Force officers how to apply close air support in a joint combat environment. It states in the FAST STICK users manual that ground actions are taking place at the same time that the air war is going on. However, the current implementation of the exercise does not display or include any type of ground action events. The only type of action involving close air support is random generation of requests for close air support (CAS) by the simulation. The only results players see is whether or not they have destroyed the CAS target. They are unable to observe the impact that their CAS allocation decision has on the outcome of a battle in a particular scenario or setting.

In the current exercise, the CAS requests generated by the simulation have no relevance to any type of concerted effort on the part of either side in the conflict to achieve some tactical land or battle objective. The importance of supporting requests for close air support is emphasized in the exercise by penalizing a team 500 points for not responding. Although this is adequate in terms of ensuring that a team responds to such requests, it does not truly emphasize to players the important role that close air support can play in a joint operation. If players were in a real world situation such as portrayed in FAST STICK, they would most likely be informed or at least aware of the ground actions occurring in the conflict. Therefore, this area of the exercise is somewhat lacking in providing a realistic setting for a Tactical Air Control Center. The need exists in this exercise for some type of mechanism to inform players of current ground actions occurring in the theater and the effect that the close air support missions they allocate will have on the outcome of battles in the theater.

Another problem with the FAST STICK program is the way data is stored. Exercise data is stored in a flat file storage structure and encoded with special numeric coding formats. This method of storage makes it very difficult for game controllers to change the data or parameters of the game in order to calibrate the exercise for different scenarios or learning objectives. The flat file structure and data encoding also results in much data redundancy with no means of ensuring data integrity and consistency. In order to alleviate the problems of the flat file structure, the game controllers needed a tool that would allow them to easily view or update the exercise data.

5 Land Simulation Enhancements

In order to make the FAST STICK exercise reflect the actual mission operations planning that will need to occur in a joint operations environment and provide a more realistic platform for generating close air support requests to the TACC, we enhanced the game by adding scripted battle scenarios containing land battle events to the exercise [2]. From analysis of this exercise, we determined that the best means of adding land play was not to create a separate land battle simulation that ran by itself, but to provide game controllers with a tool that would allow them to generate different battle scenarios so that they could emphasize different aspects of close air support in different settings. Such a tool would allow them to change scenarios as close air support concepts and doctrine were updated or modified.

In order to accomplish this, an extension was added to the current FAST STICK computer model so that land battle events could be included. The land battle scenario generation extension requires the game controller to design a particular series of ground actions that would occur in the exercise. The ground actions would consist of descriptions of battle events that could occur at a particular time in the exercise. A description of a battle event might state that a Country A unit was attacking a Country B unit with armor and heavy artillery. Follow on battle events would describe other actions occurring in the battle, the direction the battle was going, losses each side was taking, etc. This information would be displayed on the screen to players as the events occurred during the simulation run. At certain points during the battle, a request for a close air support would occur. Players would respond to requests by allocating a certain number and type of CAS aircraft to this request. The CAS mission would then be simulated. Depending upon the number of aircraft they assigned, the generation of a random probability, and whether the target was destroyed or not would determine the next sequence of battle events. The player would see the impact of his CAS allocation decision in the next series of battle events that occurred.

There were two reasons why it was decided to simulate ground actions in this particular manner. First, this method provided flexibility to the exercise in terms of allowing the game controllers to generate different settings or situations in which to teach different doctrine or aspects of close air support. Second, this method had very little impact on the rest of the simulation program, thereby, avoiding the problem of the exercise being radically different from its current state.

As stated above, the extension to the FAST STICK model basically consists of the addition of a sequences of battle events occurring at specified times. Whenever a CAS event occurs, the result of the CAS mission determines the next sequence of battle events. The following factors influence the result:

1. Number of CAS aircraft assigned to the close air support request.

2. Whether the CAS target was destroyed or not.

3. The result of a random number function generating a number between 0 and 1.

Figure 2 on the next page is a graphic representation of events in a scenario. The squares represent land battle events that will occur on a particular day of the exercise. The circles represent requests for close air support that will become CAS mission events, while the diamonds represent the possible paths the scenario can take after the CAS mission is completed. The numbers in each of the geometric shapes specifies an event number. These numbers are used to uniquely identify

each event in a scenario. Associated with each event type in the scenario diagram is a form where specific event data is recorded (See Figures 3, 4, and 5). The game controller uses the combination of the scenario diagram and event data forms to design battle scenarios to be entered into the simulation program.

The implementation of land simulation was accomplished by adding two new event types to the simulation program. These two event types represent battle events and decision events. Battle events in the simulation are text descriptions of events occurring in the ground battle scenario designed by the game controller. Decision events follow the completion of close air support missions. They are used to check the results of such missions to determine the next sequence of scenario events to be simulated. For each of these event types, Pascal routines were developed that simulate the events. The battle event Pascal routines stop execution of the simulation program, retrieve the text description of a battle event from a data file, and then display the text in a window on the screen of the Zenith 158. The student must then press the space bar for the simulation program to continue execution. The decision event routines added to the exercise examine data associated with the CAS mission previously simulated to determine the next sequence of scenario events to be simulated. In particular, the routines check the following information associated with a CAS mission:

1. The number of aircraft the student allocated for the CAS.

2. The current status of the CAS target, destroyed or not.

The decision routines use this information along with the generation of a random number to determine the next sequence of scenario events that will be simulated in the exercise. The sequence of events that follow a decision event can consist of any of the following combination of events:

1. All battle events.

2. Battle events followed by a CAS event, followed by a decision event.

3. A CAS event followed by a decision event.

6 Data Management Enhancements

Originally, we had planned replace the application dependent file structure in the FAST STICK simulation program with a commercial database management system (INGRES) by having every application flat file reference replaced with an INGRES embedded database call, but because of two constraints encountered in the current operating environment this was not feasible.

The first constraint encountered dealt with main memory and the MS DOS operating system environment. The Zenith 158 microcomputer running MS DOS has 640 kilobytes of internal memory available. Approximately 62 kilobytes of this memory is used by the MS DOS operating system. The INGRES database management system requires approximately 220 kilobytes of code to remain resident in main memory on a IBM compatible microcomputer. The FAST STICK simulation program with embedded database calls requires approximately 320 kilobytes of memory at the beginning of execution. However, the program may require additional memory as it is executing. Exactly how much is dependent on the number of missions being simulated. With this large amount

of memory being utilized, it is very possible for the simulation program to run out of memory while simulating a large number of missions.

The second constraint was the slow access time of the disk on the Zenith 158 microcomputer. The simulation program read and wrote large quantities of data to and from secondary storage (disk). Using the random access methods of the original application flat file system, the slow performance of the disk was not extremely noticeable. However, when data was accessed using the INGRES embedded database calls, the slower performance was very noticeable. Retrieval and storage of exercise data using the INGRES database management system would have resulted in poor run time performance of the simulation program. Additionally, the INGRES database management system for IBM compatible microcomputers only permitted database calls to be embedded within C programs. The FAST STICK simulation program was written in Pascal. In this situation, the Pascal simulation program would have had to call a C subroutine to access the database. This would have required additional run time for conversion of C data variables and structures to Pascal data variables and structures.

To circumvent these constraints and still use the INGRES DBMS, it was decided to continue to run the FAST STICK program with flat files, and also have a copy of the exercise data in an INGRES database. Game controllers can access, manipulate, and add data through the INGRES Query-By-Forms tool. The data in the database is then down-loaded to the flat files to run the simulation.

In order to down-load the data, a series of conversion routines were written that accessed the database through embedded calls and converted the data into the application flat file record format. These conversion routines were linked together by a driver routine which allows the game controller to select which flat files he or she wants updated from the database.

7 Conclusion

In this paper, we have described the FAST STICK exercise, the computer model that simulates the events that will occur during the exercise, some of the current weakness of the program, and our enhancements to overcome these problems. The addition of these enhancements allows the exercise to more closely reflect the real world air combat operations in a joint air-land military operational environment. In addition, the exercise can now be easily modified or tuned to meet changing learning objectives and doctrine.

Acknowledgments

This research was supported with funding from the Air Force Wargaming Center, CADRE/WG, Maxwell AFB, AL, 36112.

References

[1] *FAST STICK: A Tactical Air Forces Employment Feasibility Exercise*, Air Command and Staff College, Air University, Maxwell AFB, Alabama, 1988.

[2] Walker, Swen A., *Database Design and Land Battle Interface for the FAST STICK Exercise*, MS thesis, AFIT/GCS/ENG/88D-22, School of Engineering, Air Force Institute of Technology (AU), Wright-Patterson AFB OH, December 1988.

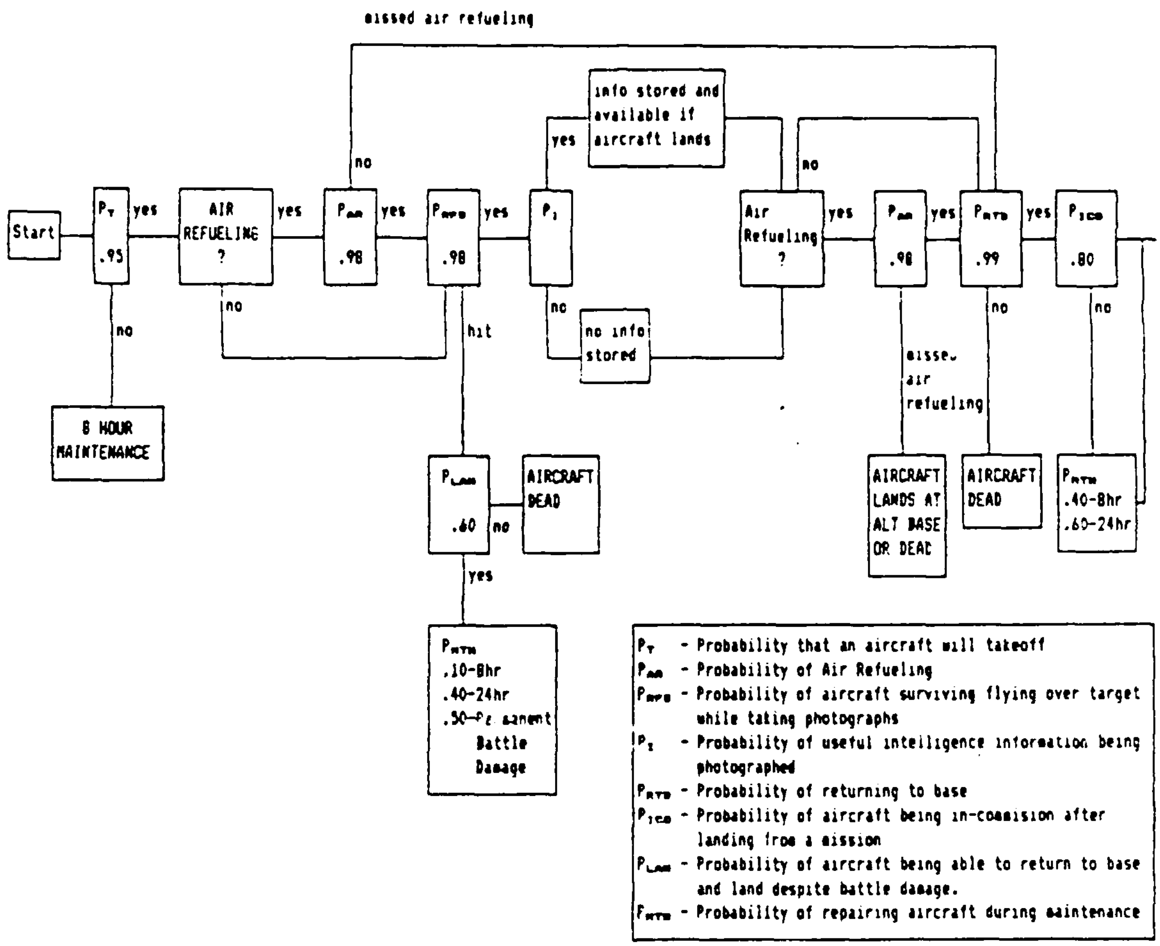

Figure 1: Reconnaissance Mission Simulation Sequence

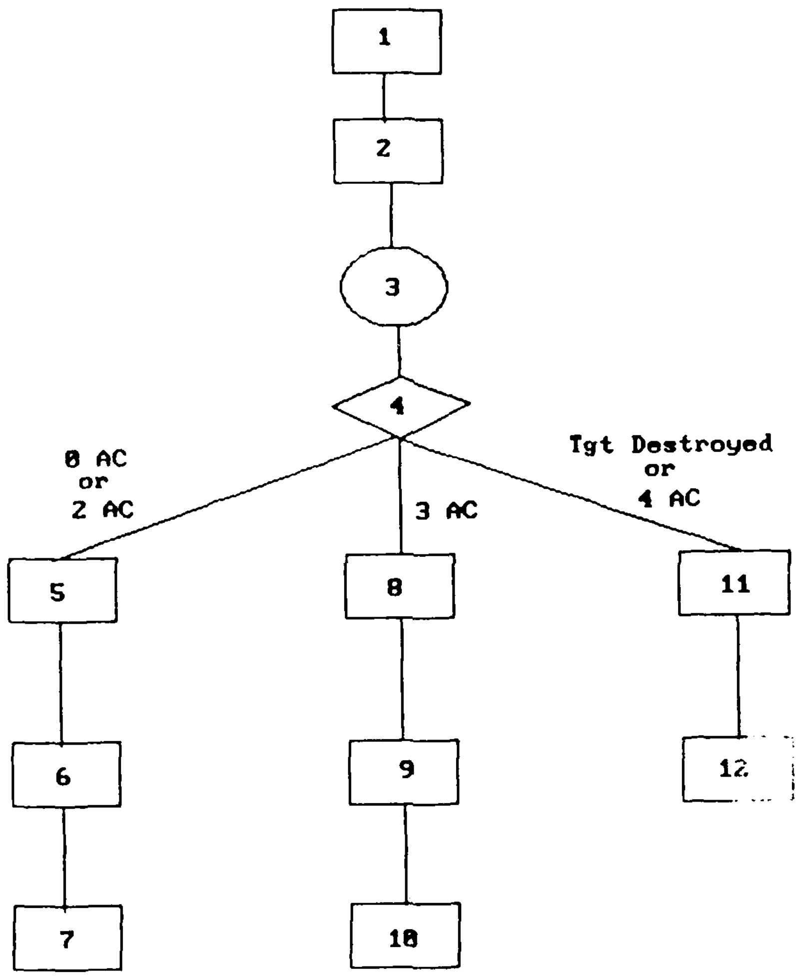

Figure 2: Graphic Representation of Scenario

Event Nbr	Game Day of Event	Game Time of Event	Description	Next Event

Figure 3: Sample Battle Event Data Form

Event Nbr	Game Day of TAR	Game Time of TAR	Max Takeoff Time for TAR Flight	TAR Target	Next Event

Figure 4: Sample TAR (CAS) Event Data Form

<table>
<tr><th>Event Nbr</th><th>Game Day of Event</th><th>Game Time of Event</th><th>TAR Event Nbr</th><th>Tgt Destroyed</th><th>AC Input</th><th>Probability</th><th>Next Event</th></tr>
<tr><td rowspan="11"></td><td rowspan="11"></td><td rowspan="11"></td><td rowspan="11"></td><td rowspan="10">NO</td><td>0</td><td>1.0</td><td></td></tr>
<tr><td rowspan="3">2</td><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td rowspan="3">3</td><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td rowspan="3">4</td><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td></td><td></td></tr>
<tr><td>YES</td><td>N/A</td><td>1.0</td><td></td></tr>
</table>

Figure 5: Sample TAR Decision Event Data Form

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE

REPORT DOCUMENTATION PAGE

Form Approved OMB No. 0704-0188

1a. REPORT SECURITY CLASSIFICATION: UNCLASSIFIED

1b. RESTRICTIVE MARKINGS: None

2a. SECURITY CLASSIFICATION AUTHORITY:

2b. DECLASSIFICATION / DOWNGRADING SCHEDULE:

3. DISTRIBUTION / AVAILABILITY OF REPORT: Approved for public release; distribution unlimited.

4. PERFORMING ORGANIZATION REPORT NUMBER(S): AFIT/EN-TR-89-2

5. MONITORING ORGANIZATION REPORT NUMBER(S):

6a. NAME OF PERFORMING ORGANIZATION: School of Engineering

6b. OFFICE SYMBOL *(If applicable)*: AFIT/ENG

7a. NAME OF MONITORING ORGANIZATION:

6c. ADDRESS *(City, State, and ZIP Code)*:
Air Force Institute of Technology
WPAFB, OH 45433-6583

7b. ADDRESS *(City, State, and ZIP Code)*:

8a. NAME OF FUNDING / SPONSORING ORGANIZATION: Air Force Wargaming Center

8b. OFFICE SYMBOL *(If applicable)*: CADRE/WG

9. PROCUREMENT INSTRUMENT IDENTIFICATION NUMBER:

8c. ADDRESS *(City, State, and ZIP Code)*: Maxwell AFB, AL 36112-5532

10. SOURCE OF FUNDING NUMBERS

PROGRAM ELEMENT NO.	PROJECT NO.	TASK NO	WORK UNIT ACCESSION NO

11. TITLE *(Include Security Classification)*: FAST STICK: A COMPUTER ASSISTED TACTICAL AIR EMPLOYMENT EXERCISE (UNCLASSIFIED)

12. PERSONAL AUTHOR(S): Swen A. Walker, Capt, USAF and Mark A. Roth, Maj, USAF

13a. TYPE OF REPORT: Technical Report

13b. TIME COVERED: FROM ______ TO ______

14. DATE OF REPORT *(Year, Month, Day)*: 5 June 1989

15. PAGE COUNT: 14

16. SUPPLEMENTARY NOTATION:

17. COSATI CODES

FIELD	GROUP	SUB-GROUP
15	06	
12	05	

18. SUBJECT TERMS *(Continue on reverse if necessary and identify by block number)*: WAR GAMES, DATA MANAGEMENT, SIMULATION, COMPUTERIZED SIMULATION

19. ABSTRACT *(Continue on reverse if necessary and identify by block number)*

FAST STICK is a computer simulated tactical air employment exercise which serves as the capstone for the theater warfare phase of the Air Force's Air Command and Staff College curriculum. Its main objective is to provide intermediate level Air Force staff officers the opportunity to apply the basic tactical employment concepts of reconnaissance, counter air, interdiction, and close air support. The Air Force Institute of Technology identified several shortcomings of the exercise and proceeded to correct them. This paper describes the exercise, the addition of a scripted scenario generator for land play, and the replacement of the application dependent file structures with a database management system.

20. DISTRIBUTION / AVAILABILITY OF ABSTRACT: ☒ UNCLASSIFIED/UNLIMITED ☐ SAME AS RPT ☐ DTIC USERS

21. ABSTRACT SECURITY CLASSIFICATION: UNCLASSIFIED

22a. NAME OF RESPONSIBLE INDIVIDUAL: Mark A. Roth, Maj, USAF

22b. TELEPHONE *(Include Area Code)*: 513-255-3576

22c. OFFICE SYMBOL: AFIT/ENG

DD Form 1473, JUN 86 *Previous editions are obsolete.*

SECURITY CLASSIFICATION OF THIS PAGE

UNCLASSIFIED